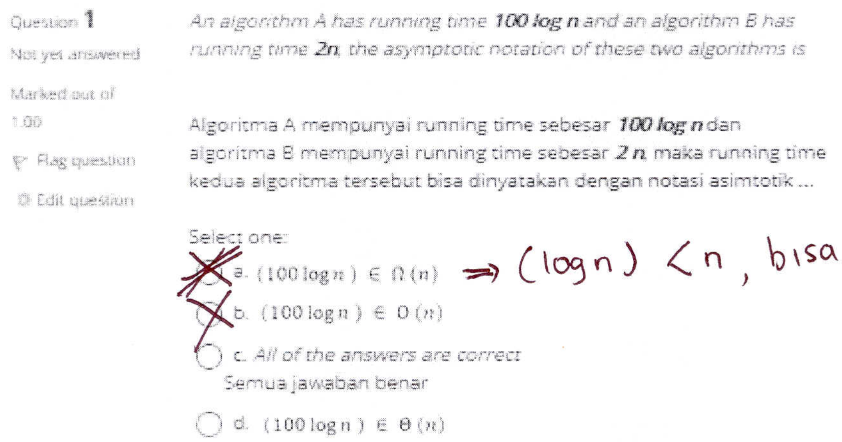
Pembahasan



Question **1**

Not yet answered

Marked out of 1.00

Flag question

Edit question

*An algorithm A has running time **100 log n** and an algorithm B has running time **2n**, the asymptotic notation of these two algorithms is*

Algoritma A mempunyai running time sebesar ***100 log n*** dan algoritma B mempunyai running time sebesar ***2 n***, maka running time kedua algoritma tersebut bisa dinyatakan dengan notasi asimtotik ...

Select one:

☒ a. $(100 \log n) \in \Omega(n)$ ⇒ (log n) < n, bisa ditulis $(100 \log n) \in O(n)$ ↓ big oh

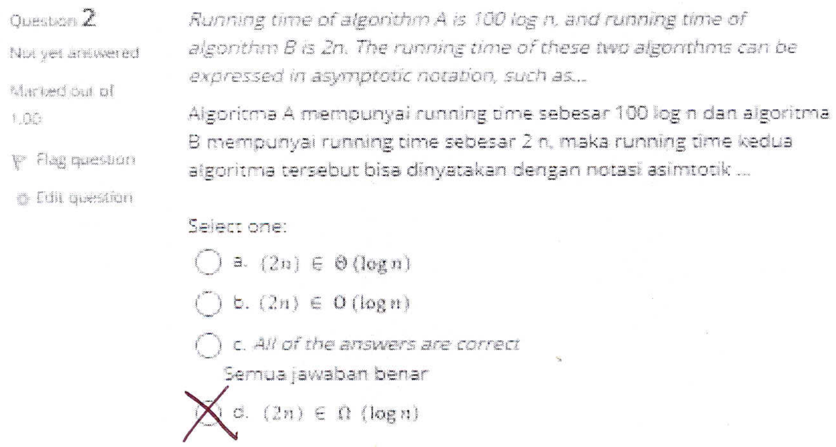
☒ b. $(100 \log n) \in O(n)$

○ c. *All of the answers are correct*
Semua jawaban benar

○ d. $(100 \log n) \in \Theta(n)$

Jawab: * notasi asimtotik ada 3 yaitu O (dibaca: big Oh), Θ (dibaca: big theta) dan Ω (big omega).

- * Suatu running time, hubungannya dg running time lain bisa dinyataka dengan salah satu dr ketiga notasi asimtotik tsbt.
- * Jika running time A < running time B, bisa ditulis (running time A) ∈ O (running time)
- * Jika running time A > running time B, bisa ditulis (running time A) ∈ Ω (")
- * Jika running time A ≈ running " , bisa ditulis (running time A) ∈ Θ (")



Question **2**

Not yet answered

Marked out of 1.00

Flag question

Edit question

Running time of algorithm A is 100 log n, and running time of algorithm B is 2n. The running time of these two algorithms can be expressed in asymptotic notation, such as...

Algoritma A mempunyai running time sebesar 100 log n dan algoritma B mempunyai running time sebesar 2 n, maka running time kedua algoritma tersebut bisa dinyatakan dengan notasi asimtotik ...

Select one:

○ a. $(2n) \in \Theta(\log n)$

○ b. $(2n) \in O(\log n)$

○ c. *All of the answers are correct*
Semua jawaban benar

☒ d. $(2n) \in \Omega(\log n)$

Jawab:

* running time A = 100 log n = kelas log n (kelas logaritmik)

* running time B = 2n = kelas n (kelas linear)

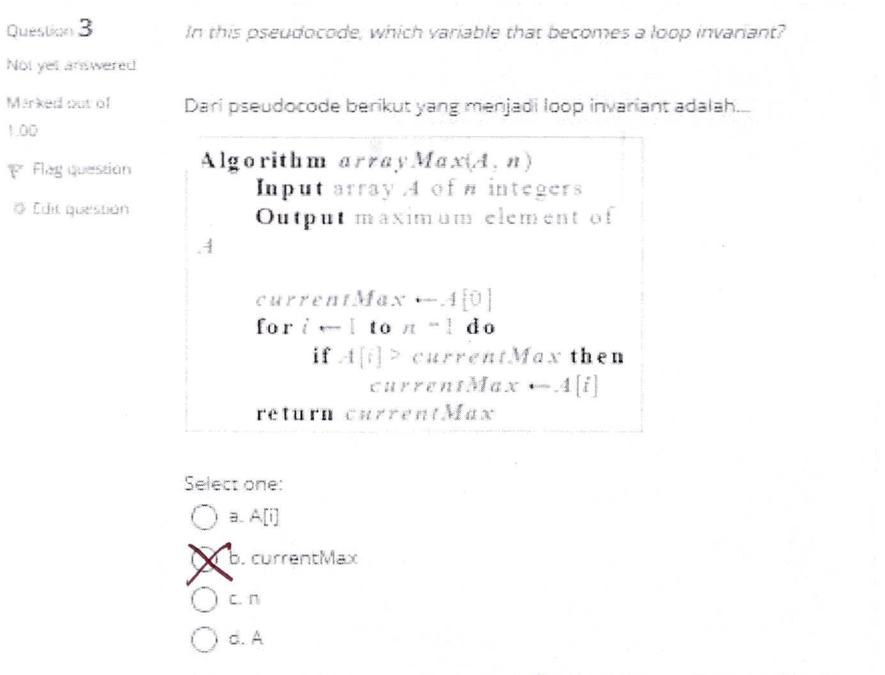
Jadi

(running time B) > (running time A)

(2n) > (log n)

sehingga bisa ditulis

$(2n) \in \Omega(\log n)$



In this pseudocode, which variable that becomes a loop invariant?

Dari pseudocode berikut yang menjadi loop invariant adalah...

```
Algorithm arrayMax(A, n)
    Input array A of n integers
    Output maximum element of
A

    currentMax ← A[0]
    for i ← 1 to n − 1 do
        if A[i] > currentMax then
            currentMax ← A[i]
    return currentMax
```

Select one:

○ a. A[i]

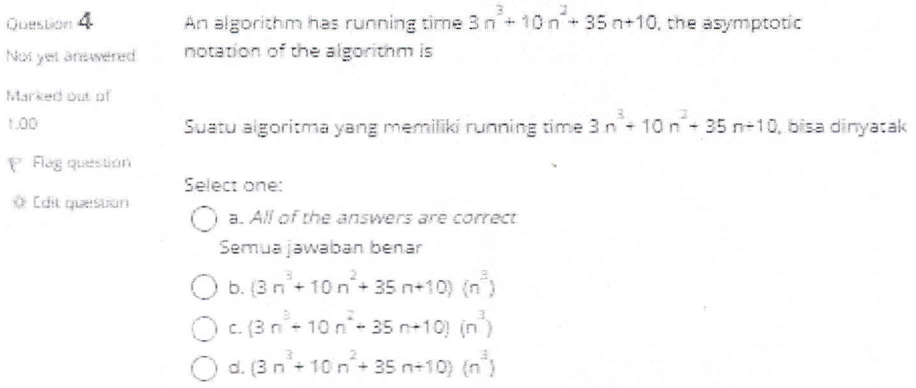
☒ b. currentMax

○ c. n

○ d. A

Jawab:

- * loop invariant adalah variabel yg menjadi kunci dalam suatu perulangan.
- * tujuan dari perulangan tsbt adalah untuk mengisi variabel current Max.
- * Shg loop invariantnya adalah variabel current Max



An algorithm has running time 3 n^3 + 10 n^2 + 35 n+10, the asymptotic notation of the algorithm is

Suatu algoritma yang memiliki running time 3 n^3 + 10 n^2 + 35 n+10, bisa dinyatak

Select one:

○ a. *All of the answers are correct*
Semua jawaban benar

○ b. (3 n^3 + 10 n^2 + 35 n+10) (n^3)

○ c. (3 n^3 + 10 n^2 + 35 n+10) (n^3)

○ d. (3 n^3 + 10 n^2 + 35 n+10) (n^3)

Jawab:

* running time algoritma A: $3n^3 + 10n^2 - 35n - 10$ = kelas n^3

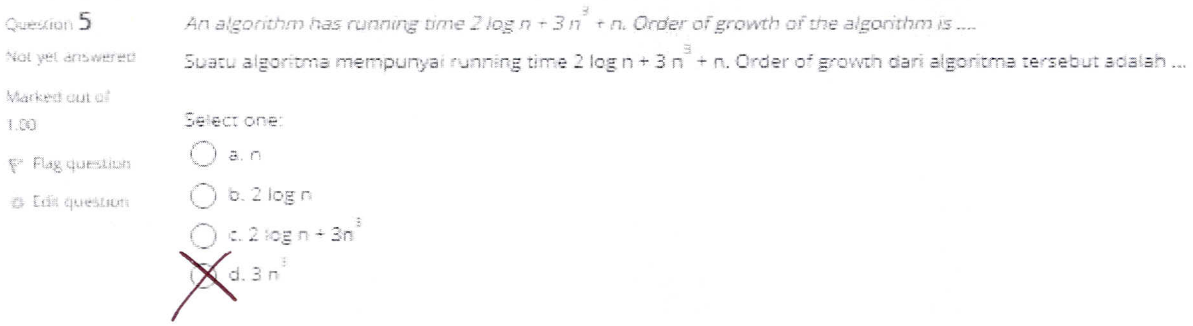
~~A to~~ running time algoritma B = n^3

* Karena algoritma A dan B berada dalam satu kelas, maka bisa dituliskan:

$(3n^3 + 10n^2 + 35n + 10)$ Ƶ (n^3)

Shg:

$(3n^3 + 10n^2 + 35n + 10) \in \Theta(n^3)$



Question 5

Not yet answered

Marked out of 1.00

Flag question

Edit question

An algorithm has running time 2 log n + 3 n^3 + n. Order of growth of the algorithm is ....

Suatu algoritma mempunyai running time 2 log n + 3 n^3 + n. Order of growth dari algoritma tersebut adalah ...

Select one:

○ a. n

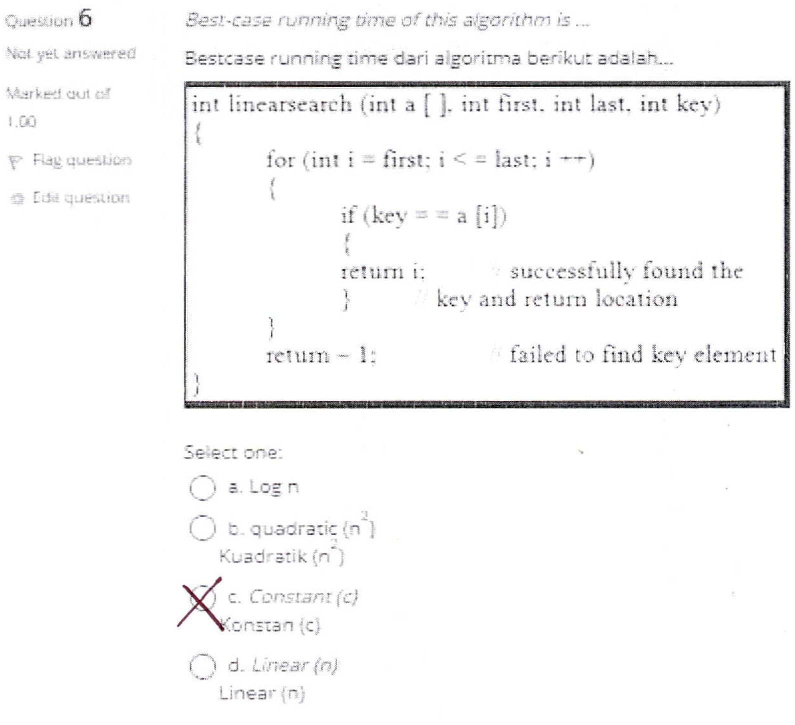
○ b. 2 log n

○ c. 2 log n + 3n^3

☒ d. 3 n^3

* 'Dari running time / persamaan 2 log n + $\underline{3n^3}$ + n → pangkat tertinggi adalah n^3,

* Maka order of growth dr persamaan tsbt adalah $\underline{3n^3}$



Question 6

Not yet answered

Marked out of 1.00

Flag question

Edit question

Best-case running time of this algorithm is ...

Bestcase running time dari algoritma berikut adalah...

```
int linearsearch (int a [ ], int first, int last, int key)
{
        for (int i = first; i < = last; i ++)
        {
                if (key = = a [i])
                {
                return i;          // successfully found the
                }          // key and return location
        }
        return - 1;             // failed to find key element
}
```

Select one:

○ a. Log n

○ b. quadratic (n^2)
Kuadratik (n^2)

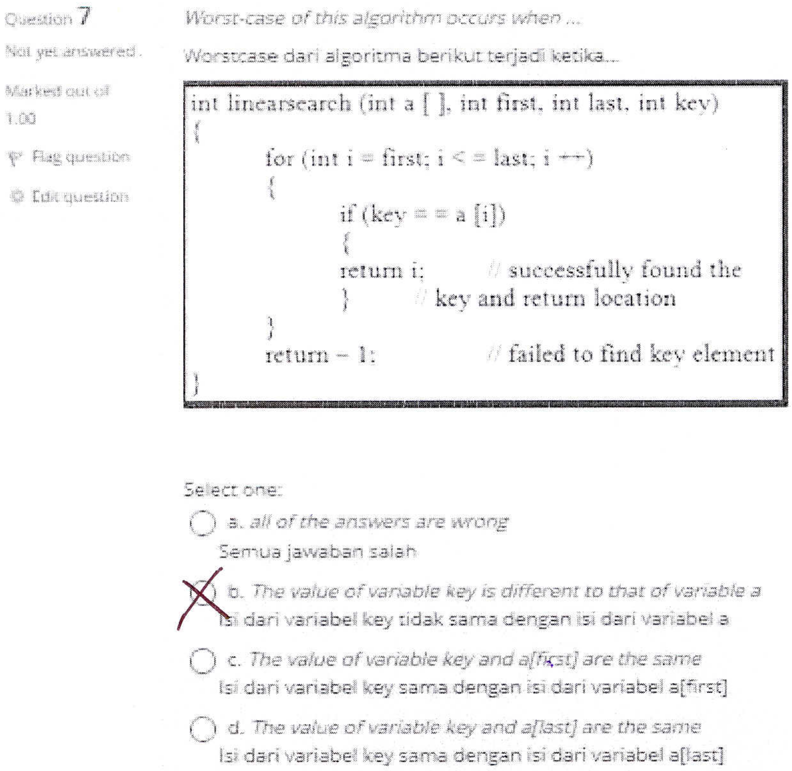
☒ c. *Constant (c)*
Konstan (c)

○ d. *Linear (n)*
Linear (n)

* Base case ~~da~~ dari algoritma tersebut terjadi ketika isi array yang terjadi berada pada index pertama, atau ketika key == a[1]

* Sehingga operasi didalam perulangan hanya dilakukan sebanyak 1 kali

* sehingga running time algoritma tsbt dlm kondisi base case adalah ~~1~~(konstan (c))



Question 7

Not yet answered

Marked out of 1.00

Flag question

Edit question

Worst-case of this algorithm occurs when ...

Worstcase dari algoritma berikut terjadi ketika...

```
int linearsearch (int a [ ], int first, int last, int key)
{
        for (int i = first; i < = last; i ++)
        {
                if (key = = a [i])
                {
                return i;           // successfully found the
                }           // key and return location
        }
        return – 1;                 // failed to find key element
}
```

Select one:

○ a. *all of the answers are wrong*
Semua jawaban salah

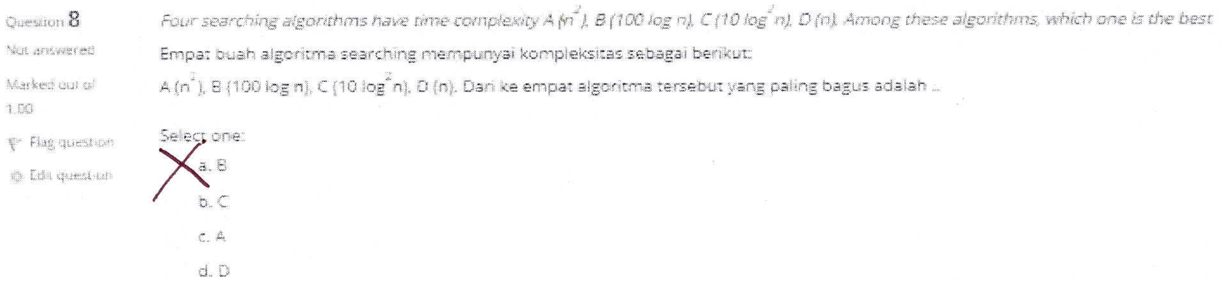
☒ b. *The value of variable key is different to that of variable a*
Isi dari variabel key tidak sama dengan isi dari variabel a

○ c. *The value of variable key and a[first] are the same*
Isi dari variabel key sama dengan isi dari variabel a[first]

○ d. *The value of variable key and a[last] are the same*
Isi dari variabel key sama dengan isi dari variabel a[last]

* worstcase running time terjadi ketika key yang di cari tidak ada di dalam a.

* sehingga looping akan terus berjalan sampai semua isi dari a diperiksa



Question 8

Not answered

Marked out of 1.00

Flag question

Edit question

Four searching algorithms have time complexity A (n^2), B (100 log n), C (10 $\log^2 n$), D (n). Among these algorithms, which one is the best

Empat buah algoritma searching mempunyai kompleksitas sebagai berikut:

A (n^2), B (100 log n), C (10 $\log^2 n$), D (n). Dari ke empat algoritma tersebut yang paling bagus adalah ...

Select one:

☒ a. B

b. C

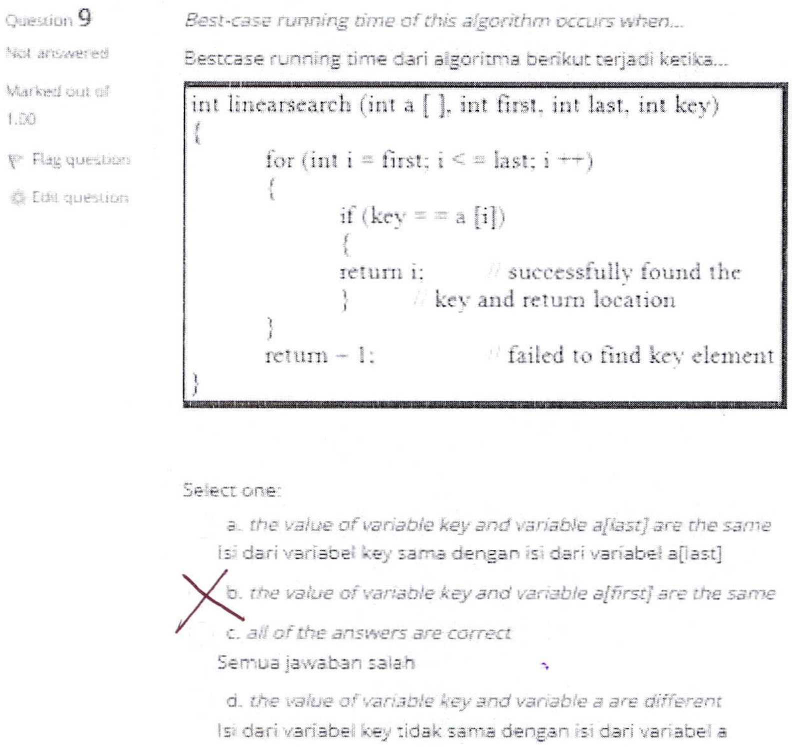
c. A

d. D

* Algoritma yang paling bagus adalah algoritma yang kompleksitasnya (running timenya) paling rendah.

* Kalau diurutkan, running time algoritma^2 tersebut adalah sebagai berikut:

$\underline{100 \log n} < 10 \log^2 n < n < n^2$



Question 9

Not answered

Marked out of 1.00

Flag question

Edit question

Best-case running time of this algorithm occurs when...

Bestcase running time dari algoritma berikut terjadi ketika...

```
int linearsearch (int a [ ], int first, int last, int key)
{
        for (int i = first; i < = last; i ++)
        {
                if (key = = a [i])
                {
                return i;        // successfully found the
                }          // key and return location
        }
        return – 1;              // failed to find key element
}
```

Select one:

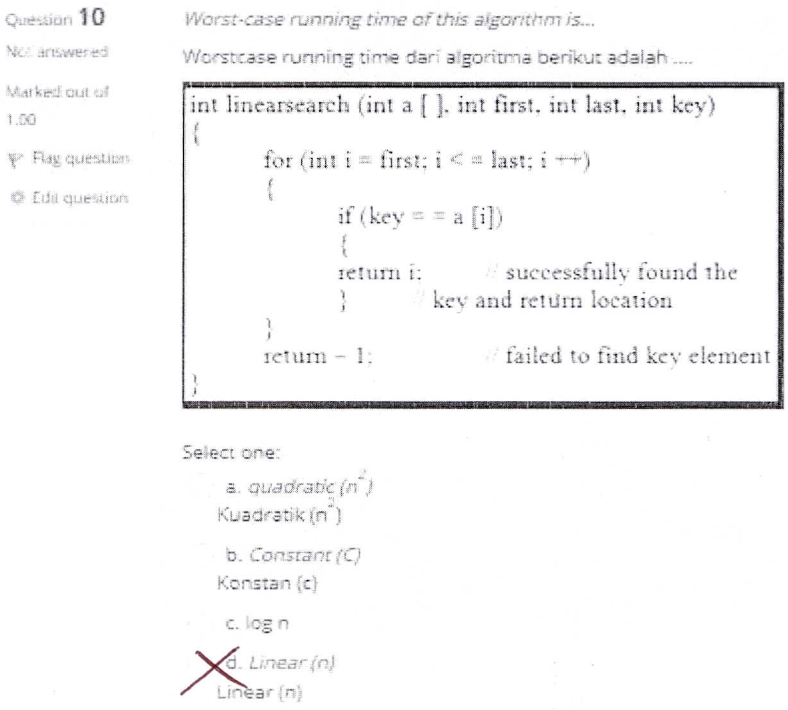
○ a. *the value of variable key and variable a[last] are the same*
Isi dari variabel key sama dengan isi dari variabel a[last]

[✓ b. *the value of variable key and variable a[first] are the same*]

○ c. *all of the answers are correct*
Semua jawaban salah

○ d. *the value of variable key and variable a are different*
Isi dari variabel key tidak sama dengan isi dari variabel a

Jawab: penjelasan sama dengan penjelasan pada no 6



Question 10

Not answered

Marked out of 1.00

Flag question

Edit question

Worst-case running time of this algorithm is...

Worstcase running time dari algoritma berikut adalah ....

```
int linearsearch (int a [ ], int first, int last, int key)
{
        for (int i = first; i < = last; i ++)
        {
                if (key = = a [i])
                {
                return i;        // successfully found the
                }          // key and return location
        }
        return – 1;              // failed to find key element
}
```

Select one:

○ a. *quadratic (n^2)*
Kuadratik (n^2)

○ b. *Constant (C)*
Konstan (c)

○ c. log n

[✓ d. *Linear (n)*
Linear (n)]

Jawab:
* worst case dari algoritma tersebut terjadi ketika key tidak ~~sa~~ ada di dalam variabel a, sehingga semua isi dari variabel a akan di cek.

* Jika isi dari variabel a sejumlah n, maka perulangan akan dilaksanakan sebanyak n kali